

HERRAMIENTAS PARA LA ENSEÑANZA DE MICROCONTROLADORES ORIENTADAS AL EEES

B. MARTÍN-DEL-BRÍO, A. BONO-NUEZ Y P. BUISÁN

Departamento de Ingeniería Electrónica y Comunicaciones. Universidad de Zaragoza. España

En el marco del Espacio Europeo de Educación Superior el aprendizaje del alumno y el desarrollo de competencias se convierten en el centro del proceso educativo. En esta línea, presentamos diversas herramientas para la enseñanza de microcontroladores, gratuitas o de muy bajo coste, aunque con prestaciones similares a los disponibles en un laboratorio universitario tradicional. Dichas herramientas están orientadas a facilitar el trabajo activo y autónomo del estudiante, de forma que pueda hacer uso de ellas a su ritmo, en su casa.

1. Introducción.

En el sistema de enseñanza tradicional la materia y el profesor han sido el centro del proceso educativo, condicionando tanto los métodos (clase magistral, prácticas de laboratorio presenciales,...) como las herramientas (pizarra, pantalla, laboratorio). Sin embargo, en el contexto del Espacio Europeo de Educación Superior (EEES) [1], el alumno y el desarrollo de sus capacidades pasan a ocupar el lugar central (“de la Universidad de enseñar, a la Universidad de aprender”). Si consideramos además la orientación más profesional que se quiere imprimir a los estudios y la necesaria adquisición de competencias transversales, todo apunta en el sentido de que métodos y herramientas deberán experimentar cambios. En el caso de la metodología, en este nuevo marco cobran sentido los denominados métodos de enseñanza activos, como puedan ser el estudio de casos o el aprendizaje basado en proyectos [2], que cada vez cuentan con mayor presencia en los estudios de ingeniería.

En el presente trabajo incidiremos más en las herramientas que en la metodología. En el mejor de los casos se dispone en la Universidad española actual de laboratorios razonablemente equipados, pero el número de horas dedicadas a las prácticas tradicionales resultan limitadas. Si se trata de potenciar el trabajo activo es necesario que el estudiante tenga un acceso sin límite al laboratorio, lo cual va a colisionar por un lado con el limitado número de horas que el profesor puede dedicar a atender presencialmente el laboratorio y, por otro, con el propio espacio físico disponible, posiblemente insuficiente en el caso de grupos medios y grandes.

Una buena solución consistiría en proporcionar al estudiante herramientas que pueda manejar por su cuenta en su casa, a su ritmo, para el desarrollo de tareas y proyectos de asignatura. Sería deseable que dichas herramientas fueran gratuitas o de coste muy bajo, pero ofreciendo prestaciones similares a las disponibles en los laboratorios presenciales. Siguiendo este razonamiento, durante los últimos años hemos desarrollado diversas herramientas software y hardware para la enseñanza de microcontroladores [3, 4, 5], en el marco de nuestra asignatura de “Microprocesadores” que impartimos en la titulación de Electrónica Industrial. En este trabajo mostraremos la actualización tecnológica de dicha asignatura, situándola ya en el contexto del EEES, favoreciendo, por ejemplo, la enseñanza autónoma y activa, así como el trabajo continuo.

El artículo queda organizado de la siguiente manera. En la Sección 2 expondremos algunas de las herramientas para microcontroladores que hemos empleado hasta ahora, junto con las que estamos desarrollando y evaluando en el momento actual. En la Sección 3 describiremos las herramientas que estamos desarrollando para la práctica con periféricos del microprocesador, uno de los aspectos fundamentales del trabajo en esta disciplina, sobre todo en el marco de una titulación de Electrónica Industrial. Finalmente, expondremos nuestras conclusiones y trabajo futuro.

2. Herramientas educativas para la enseñanza de microcontroladores.

Como es bien sabido, los microcontroladores (μC) son circuitos integrados que incluyen en un solo chip CPU, memoria volátil (RAM) y no volátil (ROM, FLASH, EPROM, EEPROM), periféricos (convertidores A/D y D/A, temporizadores PWM) y puertos de entrada/salida (paralelos y serie). Debido a su elevado nivel de integración (pequeño tamaño), coste reducido y bajo consumo, todo equipo electrónico actual incluye μC s: equipos de consumo, de comunicaciones, electrodomésticos, etc.; por ejemplo, un automóvil moderno incluye decenas de ellos, desempeñando todo tipo de tareas de control, monitorización y seguridad. Por todo ello, la enseñanza de los μC es fundamental en el contexto de la enseñanza de la electrónica [6].

En esta Sección describiremos algunas de las herramientas software y hardware que hemos encontrado útiles para la enseñanza de microcontroladores en el sentido indicado en la Sección 1: herramientas gratuitas o de coste muy reducido, pero ofreciendo prestaciones similares a las disponibles en los laboratorios presenciales, de manera que faciliten el trabajo autónomo del estudiante en su casa, a su ritmo, contribuyendo a metodologías de enseñanza activas en el marco del EEES. Algunas de estas herramientas han sido desarrolladas por nosotros y otras son comerciales.

2.1. Herramientas para el 68HC11

Hasta ahora, nuestra asignatura se ha basado en el clásico Motorola 68HC11 [7]. Para dicho μC , y ha medida de nuestra asignatura, desarrollamos hace un tiempo una serie de herramientas software y hardware, como son el simulador Visual11 y la placa denominada Kit del 68HC11 [3, 4, 5].

Por un lado, el simulador Visual11 [4] en su día tuvo una cierta repercusión y fue empleado en diversos centros de enseñanza. Pero las continuas necesidades de actualización del software debido a actualizaciones y cambios en los sistemas operativos (se desarrolló para Windows 95, pronto actualizado a Windows 98, Milenium, 2000, XP,...), implicaban un mantenimiento y actualización periódico, lo cual exige una gran dedicación por parte de los profesores. Por este motivo, hace un tiempo decidimos adoptar un software comercial que fuera gratuito o de bajo coste, y que fuese la propia compañía desarrolladora la que se dedicase a actualizaciones y mejoras. De los diversos paquetes evaluados seleccionamos el excelente simulador Shadow11 [8] (*shareware*, Fig. 1), que hemos utilizado durante los últimos años. La versión gratuita mantiene casi todas sus funcionalidades, y el coste de las licencias de la versión completamente funcional resulta extremadamente bajo (pocos euros).

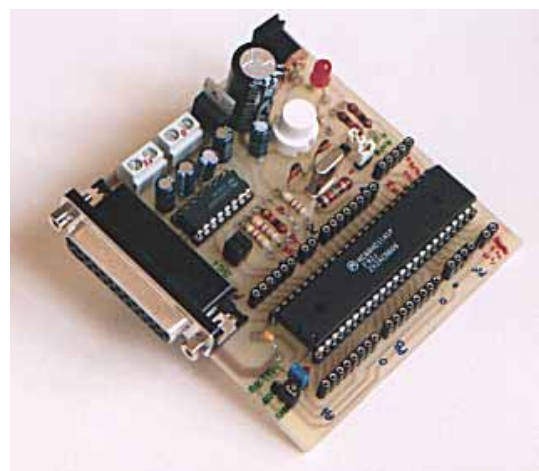
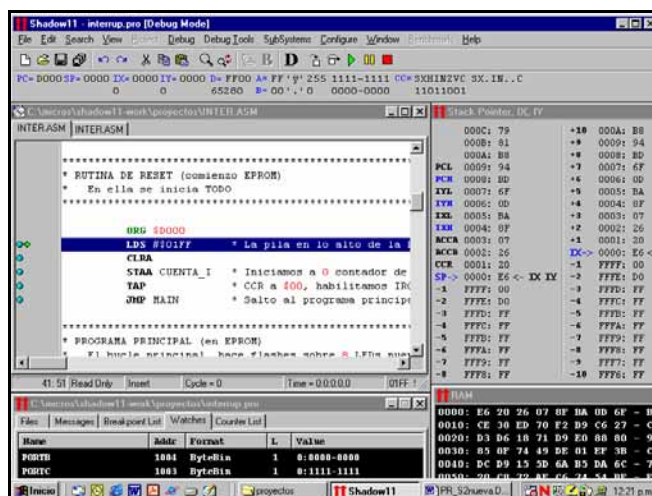


Figura 1. Entorno de desarrollo Shadow11 y Kit del 68HC11 (montado por un estudiante)

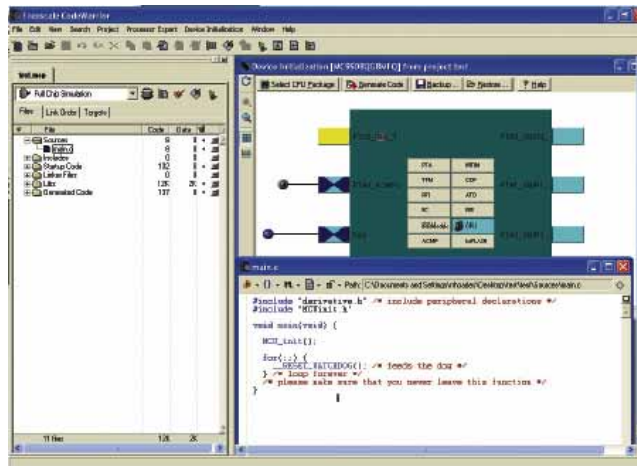


Figura 2. Entorno de desarrollo Codewarrior para los microcontroladores de Freescale [9]

En las prácticas de laboratorio presencial hemos venido utilizando la tarjeta de Motorola 68HC11EVBU, pero para su trabajo personal el alumno dispone de nuestro “Kit del 68HC11” [3, 5], cuya funcionalidad es equivalente, pero cuyo precio resulta muy reducido. El “kit” consiste básicamente en un documento en el que se detalla el montaje y puesta en marcha de una placa mínima basada en el 68HC11, que el estudiante puede él mismo construir (Fig. 1), y la cual se maneja desde cualquier computador PC compatible; el kit incluye el software necesario para ello (gratis). Aquellos estudiantes que no estén interesados en realizar un PCB y soldar, pueden montar dicho kit sobre una clásica placa de prototipos (de inserción). El coste de los componentes es de unos 24€; todos los componentes son muy comunes y pueden adquirirse en cualquier tienda o distribuidor de componentes electrónicos, incluido el microcontrolador (permite el empleo de varias versiones de la familia 68HC11, entre ellas, las más usuales y fácilmente disponibles).

2.2. Herramientas para el Freescale 68HC08

El 68HC11 es un dispositivo que Motorola/Freescale ya no promociona, cada vez es más difícil de adquirir y ni siquiera dispone de memoria *flash*. Por ello nos planteamos adaptar la asignatura a los nuevos tiempos y emplear un microcontrolador más moderno. Opciones claras son los PIC de Microchip y los AVR de Atmel, ambos de gran popularidad, tanto en la industria como en la enseñanza. La arquitectura de los AVR es particularmente moderna, pues se ha diseñado recientemente teniendo en cuenta los requisitos necesarios para que los compiladores de lenguaje C sean eficientes; además es un microcontrolador rápido, barato, de bajo consumo, y con versiones que cubren desde la gama más baja hasta la de más elevadas prestaciones. Por todo ello, en pocos años el empleo de los AVR se ha extendido enormemente.

No obstante, recientemente la división de semiconductores de Motorola se ha escindido pasando a denominarse Freescale, cuya política está siendo mucho más agresiva que la de la vieja Motorola (marca que se dedica hoy en día casi exclusivamente a comunicaciones). El 68HC08 reemplaza a los clásicos 6805 y 6811 en las aplicaciones de rango bajo-medio. Aunque el 68HC12 (de 16 bits) puede considerarse la evolución del 6811, no está disponible en encapsulados DIP, lo que dificulta el montaje de prototipos por los estudiantes; además es un dispositivo muy potente y, por lo tanto, caro.

En definitiva, la familia 68HC08 conjuga precio y prestaciones, existen versiones DIP, es de fácil adquisición, y dispone de herramientas potentes y gratuitas, como el entorno integrado Codewarrior [7, 9] (Fig. 2), gratuito, profesional, que simula ensamblador y C (hasta 16K gratis), permite grabar el programa en la memoria flash del μC y su ejecución controlada en el mismo (paso a paso, puntos de ruptura). Esta última posibilidad, la ejecución controlada de programas sobre el hardware real sin

necesidad de herramientas adicionales (como los costosos emuladores en circuito), es un punto para nosotros fundamental, y que ninguna otra casa ofrece a coste cero (que nosotros sepamos).

Por todos estos motivos, y teniendo en cuenta “el peso de nuestra historia”, en estos momentos estamos reorganizando la nueva asignatura entorno al Freescale 68HC08 (y S08), para lo cual, entre otras cosas, estamos evaluando y desarrollando diversas herramientas, que pasamos a describir.

Por un lado, en cuanto a las herramientas software, hoy en día hay disponibles entornos de desarrollo tan potentes (y además gratuitos) que pensamos que ya no tiene ningún sentido el desarrollar un simulador propio para la asignatura (como fue el caso de Visual11 en su día).

Si se desea una herramienta sencilla para su empleo por los alumnos, el simulador TExaS incluido en el texto de Valvano [10, 11] es recomendable desde un punto de vista pedagógico (es sencillo y fácil de aprender), mientras que el entorno de la empresa P&E [12] es más profesional. El entorno integrado Codewarrior (Fig. 2), desarrollado para Freescale por Metrowerks [9], y que se distribuye de forma gratuita, es probablemente el entorno más potente. Aunque al alumno le costará más su aprendizaje, pensamos que este esfuerzo merece la pena por su potencia y posibilidades: edita, compila y simula ensamblador y C (como dijimos antes, el compilador de C es gratuito hasta un tamaño de programa de 16KB, más que suficiente para trabajos y proyectos de asignatura e, incluso, de fin de carrera); también tiene la posibilidad de simular hardware. Además, Codewarrior conecta con hardware real permitiendo grabar programas en la memoria flash del μ C, así como su ejecución paso a paso y con puntos de ruptura en el propio hardware, todo ello a coste cero.

En cuanto a las herramientas hardware, en primer lugar nos planteamos que los estudiantes puedan montar sus propias placas para trabajar por su cuenta, en su casa, con el fin de que desarrollen pequeños ejercicios y trabajos de asignatura de manera autónoma y a su ritmo, sin depender de un laboratorio universitario. Para ello se requieren dispositivos con encapsulado DIP (más fáciles de montar), que resulten baratos pero con suficientes recursos integrados. El 68HC08GP32 [7] cubre las aplicaciones de prestaciones medias y altas (32KB de *flash*, y multitud de periféricos integrados en un DIP40 por unos \$6).

Entorno a este versátil dispositivo hemos desarrollado un “Kit del 08” (Fig. 3), manejado con el programa gratuito Codewarrior, en una filosofía similar a la del “Kit del 11” (Sección 2.1). De esta manera el alumno puede desarrollar programas en ensamblador o en lenguaje C (hasta 16K), simularlos, grabarlos en el HC08 e incluso ejecutar paso a paso, con puntos de ruptura, etc., gracias al pequeño programa monitor MON08 que todos los dispositivos de la familia 68HC08 integran en ROM. El precio de los componentes de un “Kit del 08” es de unos 20€, asequible para los estudiantes (más barato que la mayor parte de los libros de texto).

Este punto representa una importante ventaja frente a microcontroladores de otras casas, cuyos entornos software también gratuitos solo permiten grabar programas sobre la placa de aplicación y ordenar una ejecución continua (“run” o “go”), debiéndose realizar en esos otros casos un depurado mediante prueba y error, grabando y grabando una y otra vez la totalidad del programa. En esos casos, si se desea realizar el depurado del hardware mediante ejecución controlada de programas (paso a paso, puntos de ruptura), deben adquirirse equipos costosos, fuera del alcance del bolsillo de la mayor parte de los estudiantes.

Finalmente, existen versiones de la familia 08 más sencillas, con menos patillas y más baratas, denominadas por ello humorísticamente “PIC killers”. Estos dispositivos son ideales para que el estudiante monte en una placa de prototipos sin necesidad de hacer un PCB ni de soldar. El 68HC08QT4 (DIP8) [7] por alrededor de \$1 (comprado en grandes cantidades) integra oscilador, 4KB *flash*, 128B RAM, IC, OC, PWM, A/D... El reciente 9S08QG8 [7] es sorprendente; por \$1.10 en un

encapsulado DIP16 (hasta 12 E/S) a 16MHz y desde 1.8V, integra oscilador, 8KB *flash* y 512B RAM, timer, A/D (10 bits/8 canales), comparador analógico, SCI, SPI e I2C.

Este último dispositivo, como todos los miembros de la nueva familia 9S08, derivada de la HC08, además de ser más rápidos, integran un bloque BDC (*Background Debug Controller*) [7], el cual es un emulador y analizador lógico “on-chip”, que se maneja mediante el programa Codewarrior, disponiéndose así de un equipo con prestaciones de emulador en tiempo real (incluyendo analizador lógico) a coste cero.

Para ambos microcontroladores existen tarjetas baratas comercializadas por Freescale (Fig. 4) [7], como DEMO9S08QG8 (\$50) o DEMO68HC908QT4 (\$25), manejables desde cualquier computador PC compatible mediante Codewarrior a través de un puerto serie estándar (RS232 o USB, dependiendo de la tarjeta).

3. Placas modulares de periféricos.

Uno de los aspectos esenciales de la docencia en μC es la conexión y manejo de periféricos, como teclados, pantallas, diodos LED, sensores, accionadores, etc. En nuestras actuales prácticas de laboratorio el estudiante monta dichos periféricos sobre una placa de prototipos (de inserción), a la cual conecta también los puertos del μC mediante bandas de cable plano. La ventaja de esta aproximación es que el estudiante se tiene que enfrentar a la realidad de montar y poner en marcha dispositivos; su desventaja es el enorme tiempo que debe dedicar al montaje, teniendo que sacrificar otros aspectos formativos de mayor nivel por el limitado tiempo disponible.



Figura 3. Placa Kit del 68HC08.



Figura 4. Tarjetas de evaluación DEMO68HC08QT4 y DEMO9S08QG8 de Freescale.

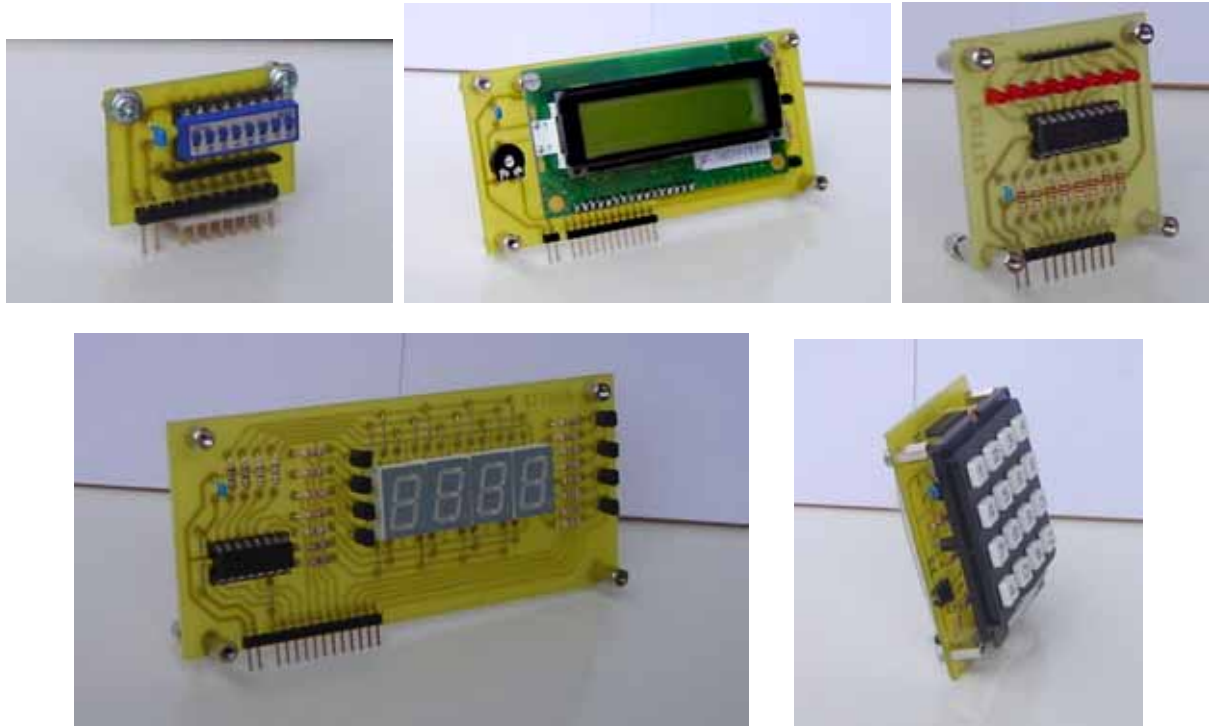


Figura 5. Ejemplos de algunas de las placas modulares de periféricos desarrolladas

Una solución a este problema consiste en proporcionar al estudiante una placa con multitud de periféricos típicos ya montados (pantalla LCD, teclado hexadecimal, pulsadores, etc.), de manera que pueda practicar con un mínimo esfuerzo de montaje (tan solo debe interconectar con cables el periférico en cuestión con el correspondiente puerto del microcontrolador). Estas placas suelen ser de gran tamaño, caras, por lo que solamente pueden estar disponibles en un laboratorio tradicional.

Para resolver este problema hemos desarrollado pequeñas tarjetas de periféricos “individuales”, modulares, que incluyen algunos de los periféricos más comunes (Fig. 5). Estas tarjetas modulares pueden emplearse en un laboratorio presencial y, además, fabricando una cantidad suficiente de dichas tarjetas, podrán ser prestadas a los estudiantes o grupos de estudiantes para su trabajo personal haciendo uso de su propia placa “kit” de microcontrolador. Como cada tarjeta incluye un solo tipo de periférico, resultan baratas; los alumnos se llevarán a casa tan solo aquellas placas concretas que necesiten en un momento dado, para posteriormente cambiarlas por otro modelo de placa para practicar con otro tipo de periférico.

Uno de los requisitos de partida que planteamos consistía en que dichas placas pudieran conectarse fácilmente a una placa de prototipos o a un zócalo, de ahí su disposición vertical (Fig. 6), o con una pequeña inclinación. De esta manera el alumno puede construir un sistema compuesto por el μC y diversas placas de periféricos casi como si de un mecano se tratase, insertando módulos e interconectando con cables. Obsérvese en las Figuras 5 y 6 que, con el fin de evitar errores, en el conector diseñado se han separado las dos patillas de alimentación (parte izquierda del conector) de las de datos y control (por lo menos ocho patillas, situadas en la parte derecha).

Algunos de los módulos desarrollados hasta el momento son los siguientes (Fig. 5):

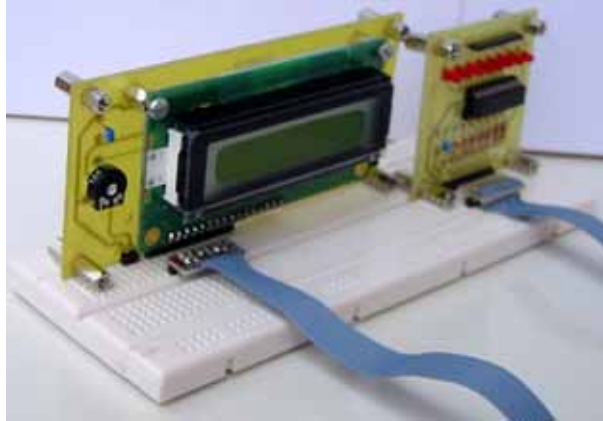


Figura 6. Conexión de placas de periféricos sobre placa de prototipos.

- 8 microinterruptores
- 8 diodos LED
- 4 pantallas de siete segmentos
- Pantalla LCD
- Teclado hexadecimal
- Sensor de temperatura, potenciómetro y zumbador

En el futuro planeamos el desarrollo de otros módulos, que incluyan motores paso a paso, sensores diversos (temperatura, humedad,...), etc.

4. Discusión y conclusiones.

De cara a la actualización tecnológica y adecuación al EEES de nuestra asignatura “Microprocesadores”, dentro del actual plan de estudios de Electrónica Industrial, hemos expuesto algunas consideraciones referentes al trabajo personal y autónomo del estudiante, de manera que se favorezca un trabajo práctico activo y continuo.

Por un lado, hemos valorado y seleccionado diversas herramientas, tanto software como hardware, para el desarrollo de trabajos y proyectos de asignatura, con los requisitos de que sean gratuitas o de coste muy reducido, de manera que el estudiante pueda descargarlas (software), adquirirlas o montarlas él mismo (hardware), para que cada estudiante o grupo de ellos pueda disponer de sus propias herramientas libremente y trabajar a su ritmo y fuera de las limitaciones de los laboratorios universitarios.

Obsérvese que hemos incidido especialmente en poder disponer de herramientas hardware que permitan el trabajo con circuitería real, pues pensamos que la simulación no es suficiente, y que el alumno debe enfrentarse a la realidad de la conexión del microcontrolador con periféricos, con sus características circuitales, temporales, etc.

Además, hemos comprobado a lo largo de los años lo estimulante y motivante que resulta para los estudiantes el sentirse capaces de montar y controlar dispositivos reales. A modo de ejemplo, no es lo mismo simular un programa que desplaza unos y ceros en la pantalla de un ordenador, que el mismo programa llevado a un hardware real con el resultado de conseguir realizar juegos de luces sobre diodos LED. Tampoco es lo mismo simular un programa que copia códigos binarios, que el mismo programa cargado en la memoria de un microcontrolador y visualizando sus efectos (representación de dígitos o mensajes de texto) en pantallas de siete segmentos o de cristal líquido.

La motivación que las prácticas hardware despierta en nuestros estudiantes de Electrónica Industrial la comprobamos día a día, año a año, tanto a través de las impresiones que extraemos del contacto directo con ellos en el laboratorio, como a partir de las encuestas oficiales y propias que realizamos: en ellas, uno de los puntos fuertes de la asignatura, muy destacado por los estudiantes, es precisamente las prácticas hardware y su conexión con la realidad.

En estos momentos estamos adaptando la asignatura en el sentido indicado; esperamos presentar próximamente los resultados del proceso de implantación y puesta en marcha, en el que se incluirá un estudio del impacto de estos cambios en el trabajo, rendimiento y motivación de los estudiantes a través de estadísticas y encuestas.

Agradecimientos.

Queremos expresar nuestro agradecimiento al Ministerio de Educación y Ciencia por su apoyo parcial a la presentación de este trabajo a través de la red temática del Capítulo Español de la Sociedad de la Educación del IEEE (TSI2005-24068-E)

Referencias.

- [1] *Declaración de Bolonia (Joint Declaration of the European Ministres of Education)*. Comisión Europea (1999). http://ec.europa.eu/education/policies/educ/bologna/bologna_en.html.
- [2] E. de los Reyes. *Introducción al estudio de casos como método de enseñanza*. Publicaciones del Instituto de Ciencias de la Educación. Prensas Universitarias de Zaragoza (2005).
- [3] B. Martín, C. Bernal. *Visual11 y Kit11: Herramientas para el aprendizaje del 6811*. III Congreso de Tecnologías Aplicadas a la Enseñanza de la Electrónica TAEE (1998)
- [4] B. Martín, C. Bernal. *A software tool for teaching microcontroller system principles*. International Journal of Electrical Engineering Education, 36, 4, 279-286 (1999).
- [5] <http://www.unizar.es/euitiz/micros.htm>
- [6] B Martín del Brío. *Sistemas Electrónicos basados en Microprocesadores y Microcontroladores*. Prensas Universitarias de Zaragoza (1999).
- [7] <http://www.freescale.com>
- [8] <http://www.geocities.com/SiliconValley/Peaks/4125/>
- [9] <http://www.metrowerks.com/>
- [10] J. W. Valvano. *Embedded Microcomputer Systems: Real Time Interfacing*. Brooks Cole (2000).
- [11] <http://www.ece.utexas.edu/~valvano/>
- [12] <http://www.pemicro.com/>